

## Contents

|  |    |
|--|----|
| EXECUTIVE SUMMARY .....                                  | 1  |
| USECASE BACKGROUND.....                                  | 1  |
| Goal.....  | 2  |
| OBJECTIVE .....  | 2  |
| BUILDING THE Directory Synchronization USE CASE.....     | 2  |
| Prerequisites .....                                      | 2  |
| Creating Connections.....                                | 2  |
| Creating the Directory Synchronization Integration ..... | 2  |
| Schedule Parameters .....                                | 4  |
| Function Call (Action).....                              | 4  |
| getNewHireATOMFeed (HCM Cloud Adapter) .....             | 11 |
| Map to getNewHireATOMFeed .....                          | 14 |
| countOfNewHires .....                                    | 15 |
| Switch Action .....                                      | 16 |
| Add Stop Action .....                                    | 19 |
| StageFileRef (Assign Activity).....                      | 20 |
| ForEachEntry.....  | 21 |
| getEmployeeDetails (REST Adapter).....                   | 23 |
| Map to getEmployeeDetails .....                          | 26 |
| writeNewUserRecord to Staging.....                       | 28 |
| Map to WriteRecordToStage .....                          | 32 |
| WriteStageFileToFTP (FTP Adapter).....                   | 34 |
| Map to WriteStageFileToFTP .....                         | 35 |
| assignATOMLRDT.....                                      | 36 |
| Tracking.....  | 36 |
| Activate and Run the Integration.....                    | 36 |
| Validating Extracted File presence on FTP Server.....    | 38 |

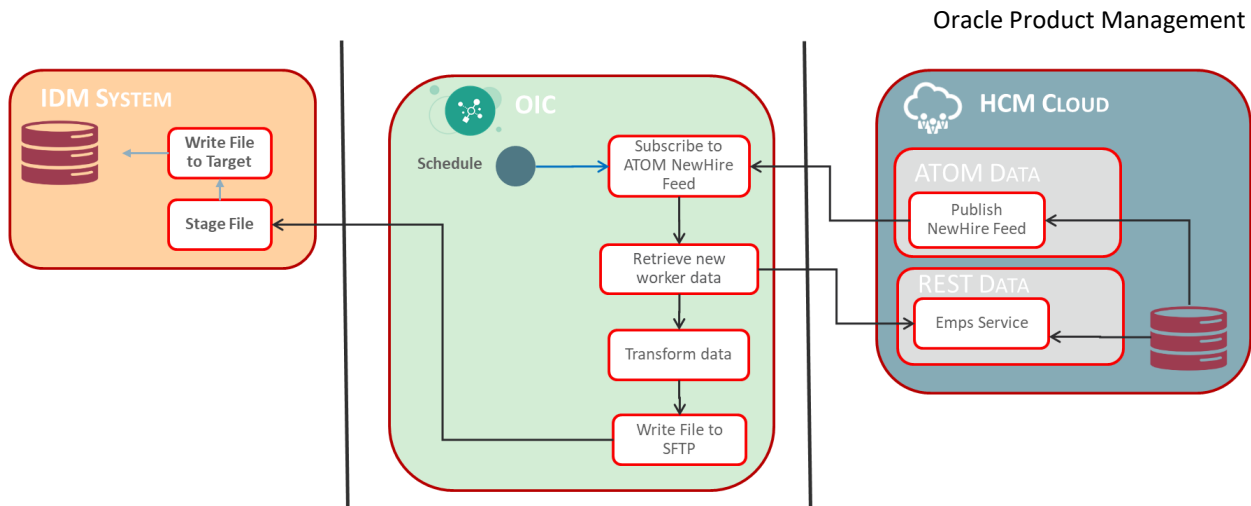
## EXECUTIVE SUMMARY

Oracle Integration provides native connectivity to Oracle and non-Oracle Software as a Service (SaaS) and On-premises applications, such as Oracle ERP Cloud, Oracle Service Cloud, HCM Cloud, Salesforce.com, Workday, EBS, SAP, NetSuite and so on. Oracle Integration adapters simplify connectivity by handling the underlying complexities of connecting to applications using industry-wide best practices. You only need to create a connection that provides minimal connectivity information for each system.

## USECASE BACKGROUND

This use case explores the use of OIC with HCM REST and ATOM services in order to satisfy the downstream needs of an Identity Management System hosted by a customer.

The following diagram illustrates the proposed interaction between the systems involved in this use case.



## Goal

This document walks you through the steps needed to replicate in your own environment what has been released in the demo title: *Directory Synchronization*

## OBJECTIVE

This document walks you through the steps needed to replicate this use case in your environment

## BUILDING THE Directory Synchronization USE CASE

This section works through the steps that are required to build the integration from scratch.

### Prerequisites

You will need access to the following applications and artifacts:

- Oracle Integration (OIC)
- HCM Cloud R13+

Note: For ClassID & StudentId: Please check with the instructor.

### Creating Connections

The following Connections have been created and configured. You will be using these connections for creating Integration flows

| Connection Name     | Connection Type |
|---------------------|-----------------|
| HCM Conn 96 06      | HCM Adapter     |
| HCM REST Conn 96 06 | REST Adapter    |
| FTP Conn 96 06      | FTP Adapter     |

### Creating the Directory Synchronization Integration


1. On the Oracle Integration home page, click **Integrations**.
2. On the Integrations page, click **Create**. The Create Integration - Select a Style/Pattern dialog is displayed.

3. Select **Scheduled Orchestration** type of integration. The **Create New** Integration dialog is displayed.
4. Enter the following information:

| Field Element   | Value   |
|---|---|
| <b>What do you want to call your integration?</b>     | Directory Synchronization <ClassId> <StudentId><br>Ex: Directory Synchronization 96 06  |
| <b>Identifier</b>                                     | Accept the default identifier value. The identifier is the same as the integration name you provided, but in upper case.  |
| <b>Version</b>  | Accept the default version number of 01.00.0000. Or, if you want to change the version number, enter the version using numbers only in this format: xx.xx.xxxx. |
| <b>What does this integration do?</b>                 | This integration demonstrates the use of the HCM Cloud Adapter along with a REST Adapter generating a file for use with an Identity Management Solution.        |
| <b>Which package does this integration belong to?</b> | <i>Leave blank</i>  |

Create New Integration

?



## Create New Integration

Enter information that describes this integration.

**Describe this integration** Use a meaningful name and description that will help others find and understand this integration. The Identifier and Version can be set only when the integration is created. The combination of Identifier and Version must be unique.

\* What do you want to call your integration?

\* Identifier

\* Version

**What does this integration do?**

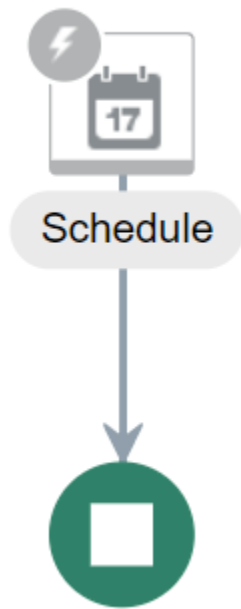
This integration demonstrates the use of the HCM Cloud Adapter along with a REST Adapter generating a file for use with an Identity Management Solution.

**Which package does this integration belong to?**

Create

Cancel

5. Click **Create**. The integration canvas is displayed.



### Schedule Parameters

The first step in defining the integration is to retrieve from **Oracle Integration** the last time the integration polled the ATOM service. This is known as the Last Run Date and Time. This allows you to avoid pulling duplicate records from the ATOM feed.

1. Click the **Schedule** icon, then select **Edit**. The **Schedule Parameters** configuration page will be displayed.
2. On the empty **Parameter Name** row, click the **+** icon.
3. Enter ATOMLastRunDateTime as the **parameter name**,
4. Enter Last successful processed ATOM pull as the optional **Description**.
5. Enter 2019-01-01T00:00:00.000Z in the **Value** column. This is to set the initial period of time to retrieve the ATOM feed from HCM Cloud to the 1<sup>st</sup> of January 2019. You may need to adjust this to a closer DateTime value depending on when you are building this integration. After the first time the integration is run, this value will be updated to reflect the actual Date and Time values.
6. Click **Close**. This will return you to the integration canvas.
7. Click on Save to save your integration flow.

### Function Call (Action)

The next step within the integration is used to set the current Date and Time of when the current ATOM poll is occurring. This will then be used as the ATOMLastRunDateTime variable the next time the integration is scheduled.

1. In the right navigation pane, click **Actions**
2. Drag the **Function Call** icon to the **+** sign following the **Schedule** label.
3. Enter getTimestamp for the function call action when prompted and Click **Create**. This will open the **Function Call** configuration dialogue

## Create Action

Function

Please give a unique name and description to this action.

\* Name

getTimestamp

Description

Enter description

Create

Cancel

- Click the **+Function** button.

Function Call action getTimestamp was created successfully.

getTimestamp Directory Synchronization 96 06 (1.0)

Function

To begin, select a function

+ Function

Then configure input parameters by adding an expression






fx

- The **Select a Function** dialogue window appears.

Integrating Your HCM with Oracle Integration

## Functions

Select a Function

|   |   |                                       |
|---|---|---------------------------------------|
|    | <b>addTime</b><br>Extension Library: DateTimeLib (1.0) <span>File: addTime1.js</span>                             |                                       |
|    | <b>addTime</b><br>Extension Library: DateTimeLib_4 (1.0) <span>File: addTime1.js</span>                           |                                       |
|    | <b>addTime</b><br>Extension Library: DateTimeLib4 (1.0)<br>Signature: dt addTime (ts,z)<br>Annotations:           | <input type="button" value="Select"/> |
|    | <b>concatMessage</b><br>Extension Library: StringUtilLibrary (1.0) <span>File: concat.js</span>                   |                                       |
|  | <b>daysOutOfWarranty</b><br>Extension Library: Days Out of Warranty (1.0) <span>File: daysOutOfWarranty.js</span> |                                       |

*Note: depending on your OIC instance, the available libraries available will differ from those shown above. If there isn't **addTime** function Register the **DateTimeLib4\_1.0.jar** navigating to the library section of OIC.*

- Find and **Click** addTime (Refer above screenshot)
- Click the **Select** button in the function's row. The configuration page for the addTime function is displayed. It shows the details of the selected function including the input and output parameters.

## Function

Function: addTime Replace

Extension Library: DateTimeLib4 - 1.0

Output Parameter: dt (String)

Output Description: No description.

## Input Parameters

Specify input for this function by adding an expression

| Parameter | Description     | Type   | Value             |
|-----------|-----------------|--------|-------------------|
| ts        | No description. | String | Add an expression |
| z         | No description. | String | Add an expression |

8. Click the pencil icon in the **Value** column for the ts parameter to access the **Expression Builder**.

ts

Expression in "getTimestamp - Directory Synchronization 96 06 (1.0)"

Inputs

View

Filter

Detach

Source

Find...

<> \*schedule

<> \*startTime

\$ATOMLastRunDateTime

\$tracking\_var\_1

\$tracking\_var\_2

\$tracking\_var\_3

Expression

Drag and drop, shuttle, or type here to build the expression

Expression Summary

9. Enter the following text directly into the **Expression** text box

```
concat(substring-before(/nssrcmpr:schedule/nssrcmpr:startTime, "."), ".000Z")
```

10. Click **Validate** in the title bar to validate the parameter.  
11. Click **Close**

 Function: addTime 





Extension Library: DateTimeLib4 - 1.0

Output Parameter: dt (String)


Output Description: No description.

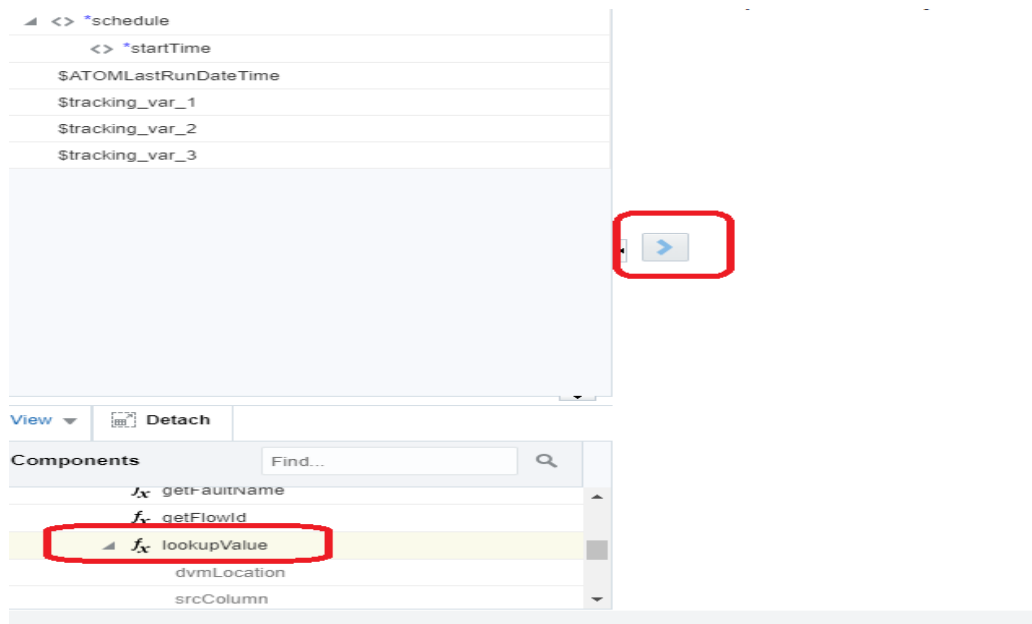
## Input Parameters

Specify input for this function by adding an expression

| Parameter | Description     | Type   | Value  |
|-----------|-----------------|--|--|
| ts        | No description. |  String |  concat( substring-before( startTime, "." ), ".000Z") |
| z         | No description. |  String |  Add an expression                                    |

You will be returned to the **Function Call** page.

12. Click the pencil icon  in the **Value** column for the z parameter to access the **Expression Builder**.
13. Select the lookupValue element which is under **Components>Functions>Integration Cloud** and click on Move button.





The **Build Lookup Function** configuration page is displayed.

The screenshot shows the 'Select Lookup Table' step of the 'Build Lookup Function' wizard. The left sidebar has a 'Lookup' tab selected. The main area displays a table with two rows: 'SystemInfo' and 'SchedulerLookup'. Both rows indicate 'COLUMNS: 2'. The 'SchedulerLookup' row is highlighted. Navigation buttons at the top include '< Back', 'Next >', 'Cancel', and 'Done'.

| Lookup | Columns         | Default    | Summary |
|--------|-----------------|------------|---------|
|        | SystemInfo      | COLUMNS: 2 |         |
|        | SchedulerLookup | COLUMNS: 2 |         |

14. Find and select the **SchedulerLookup** Lookup Table.

15. Click **Next**. The **Select Columns** configuration page is displayed.

The screenshot shows the 'Select Columns' step of the 'Build Lookup Function' wizard. The left sidebar has a 'Columns' tab selected. The main area displays a table with two columns: 'Source' and 'Target'. The 'Source' column has a button 'Select Source Column' and the 'Target' column has a button 'Select Target Column'. Below the table, it says 'No data to display'. Navigation buttons at the top include '< Back', 'Next >', 'Cancel', and 'Done'.

| Source               | Target               |
|----------------------|----------------------|
| Select Source Column | Select Target Column |
| No data to display   |                      |

16. Click on Select Source Column, select A and click on Select Target Column, select B

### Build Lookup Function

**Select Columns**  
The Lookup function requires one source and one target column.  
Click on the column headers to select from a list of available columns for this Lookup Table. Preview data from the table will be displayed.

**SchedulerLookup**

| Source     | Target               |
|------------|----------------------|
| A          | Select Target Column |
| duration   |                      |
| emailalias |                      |
| pagesize   |                      |

**Select Target Column**  
A  
B

Lookup  
Columns  
Default  
Summary

17. Click **Next**. The **Default configuration** page is displayed, **Default value 1**

18. Click **Next**. The **Summary** configuration page is displayed.

### Build Lookup Function

**Summary**  
On this page, you can confirm the resulting expression.  
Note the "Source Value" contains a placeholder value of "srcValue". This value needs to be selected in the main editor after clicking Done.

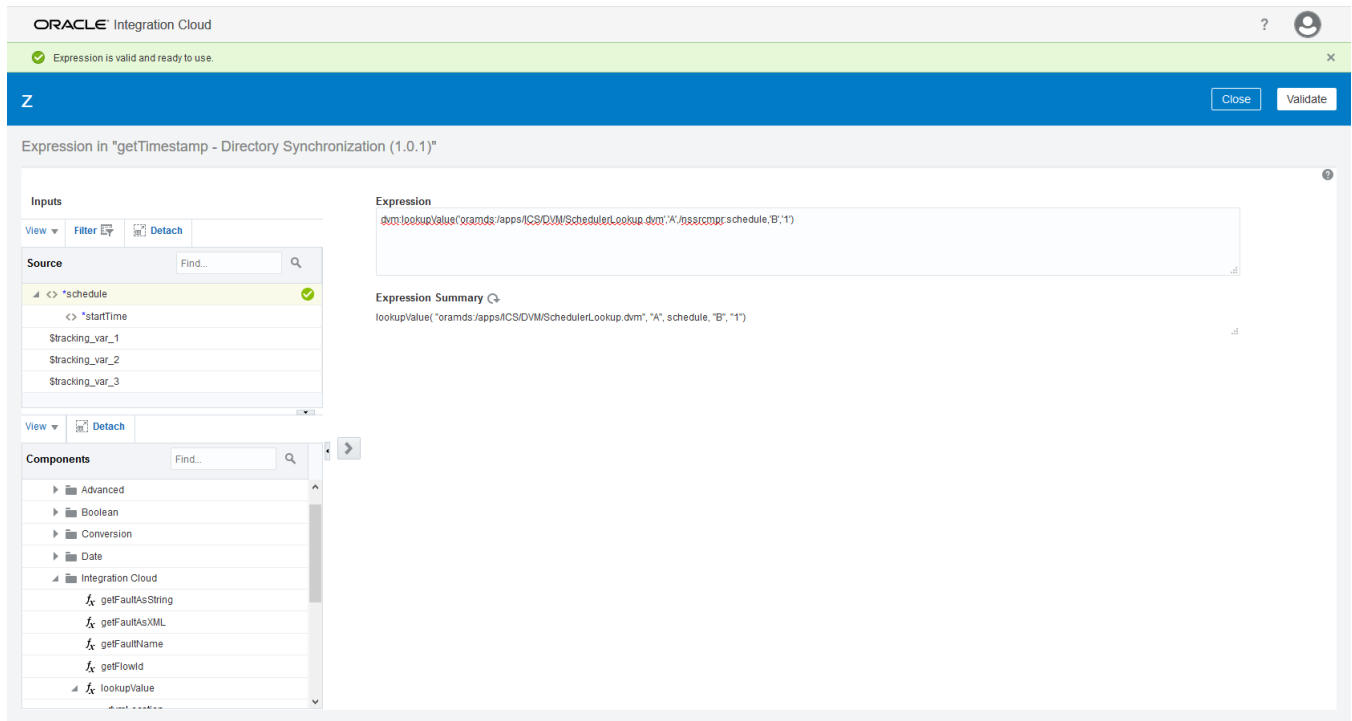
**SchedulerLookup**

| Parameter     | Value   |
|---------------|---|
| Lookup Table  | SchedulerLookup ✓   |
| Source Column | A ✓   |
| Source Value  | Note: The value for this parameter should be set in the editor. |
| Target Column | B ✓   |
| Default Value | 1 ✓   |

**Resulting Expression**  
dvm.lookupValue('oramds:/apps/ICS/DVM/SchedulerLookup.dvm','A',srcValue,'B','1')

Lookup  
Columns  
Default  
Summary

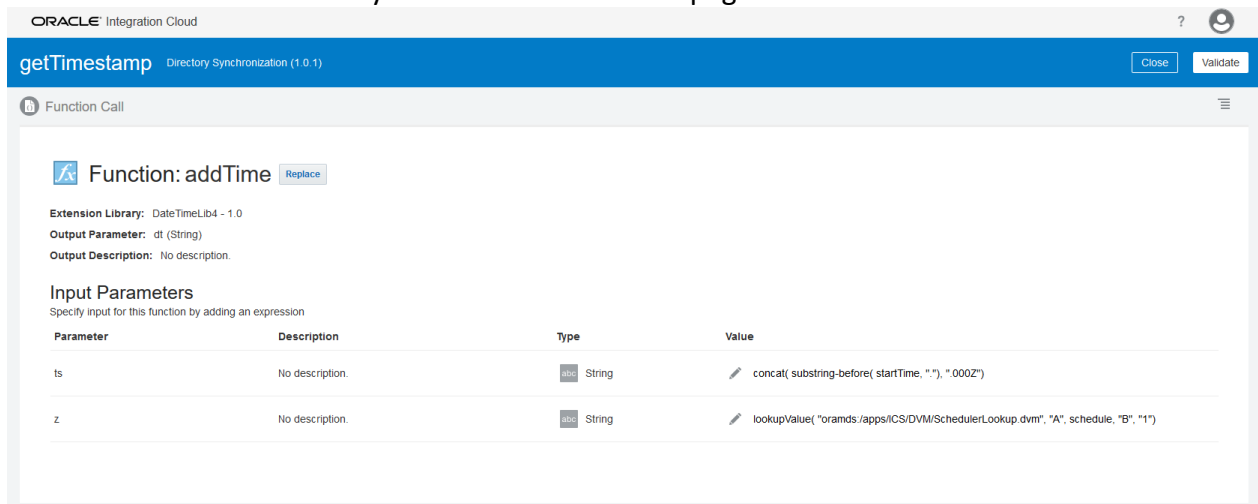
19. Click **Done**. This will return you to the **Expression Builder** page.



20. In the **Expression** text box, replace the SrcValue with the “schedule” parameter by dragging from the Source menu.

21. Click **Validate** in the title bar to validate the parameter.

22. Click **Close**. This will return you to the **Function Call** page.



23. Click **Validate** in the title bar to validate the **Function Call**.

24. Click **Close**. This will return you to the integration canvas.

### getNewHireATOMFeed (HCM Cloud Adapter)

The next task in the integration is the retrieval of the HCM Cloud ATOM feed accessed via the **HCM Cloud Adapter**.

1. In the right navigation pane, click **Invokes**.

2. Click **Oracle HCM Cloud**, then drag the HCM Conn 96 06 to the + sign following the GetTimestamp action. The **Oracle HCM Cloud Adapter** configuration wizard is displayed.
3. Enter getNewHireATOMFeed in the **What do you want to call your endpoint?**
4. Click **Next**. This will display **Actions** tab.
5. Select the radial button for **Subscribe to Updates (via ATOM Feed)**.

Configure Oracle HCM Cloud Endpoint

Welcome to the Oracle HCM Cloud Endpoint Configuration Wizard  
This wizard helps you select one of the capabilities of Oracle HCM Adapter.

**Basic Info**  
**Actions**  
Operations  
Summary

**What would you like to do with Oracle HCM Cloud Adapter?**

- ☐ **Query, Create, Update or Delete Information**  
Query business objects such as employee records etc or perform operations for employee onboarding, data sync etc.
- ☐ **Extract Bulk Data using HCM Extracts**  
Receive several records as data file on payroll records, timesheet etc
- ☒ **Subscribe to Updates (via ATOM Feed)**  
Receive latest updates since a specific date on new hires, jobs etc.
- ☐ **File Upload to WebCenter (UCM)**  
The service is a controlled check-in to Content Server.

6. Click **Next**. This will display the **Operations** tab.
7. Select the Employee New Hire as the **Select an ATOM Feed** value.
8. Select in the drop down **Max entries to process** to 250. This will ensure that most customers have all of their new hires returned in the ATOM Feed response.

Configure Oracle HCM Cloud Endpoint

Help ▾ < Back Next > Cancel Done

Configure the Operations to Perform in the Target Oracle HCM Application  
Select the business object or service and operation to use for the target integration

Basic Info  
Actions  
**Operations**  
Summary

Select an Atom Feed

Filter by atom feed

- Employee Assignment
- Employee New Hire**
- Employee Termination
- Employee Update
- Grade
- Job
- Location
- Organization
- Position

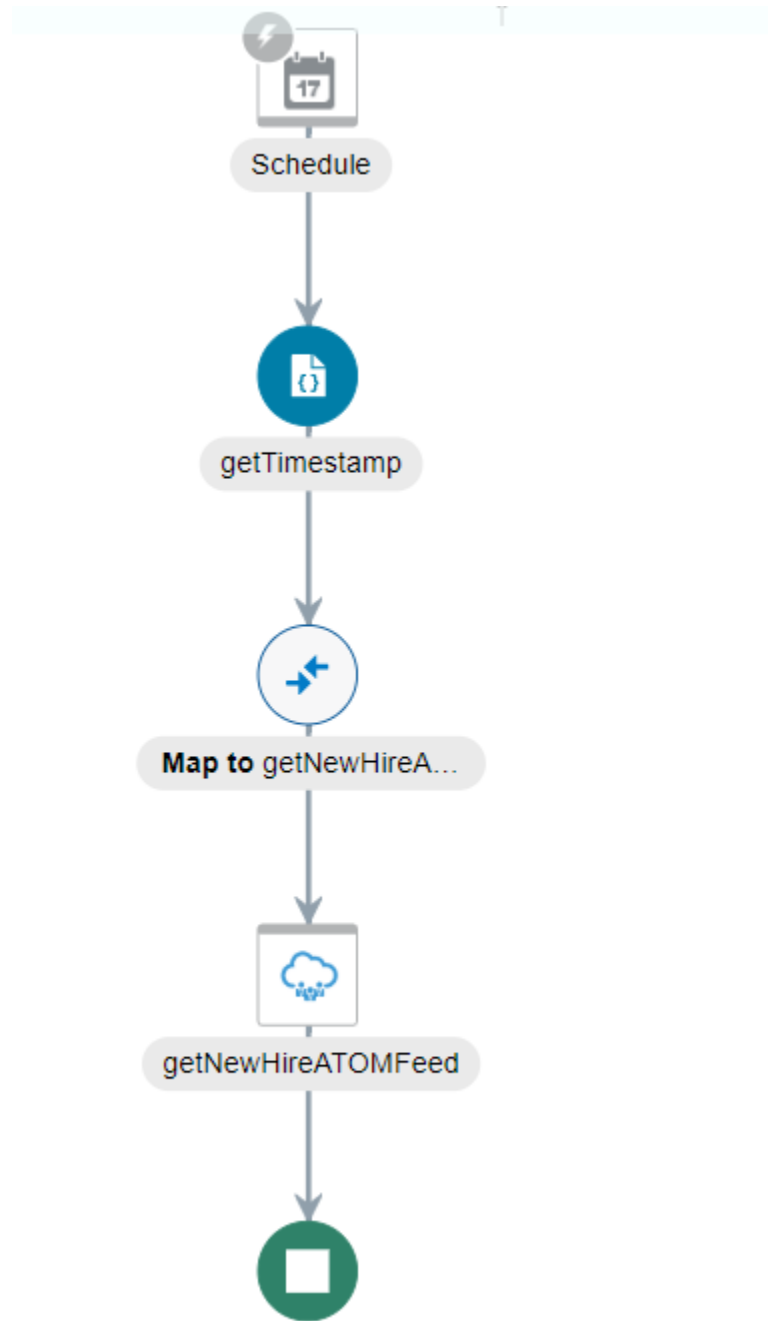
[Learn more about HCM Cloud ATOM feeds](#)

Max entries to process  
250 ▾

☐ Process Future Dated Entries Immediately

☐ Include Business Object in ATOM feeds

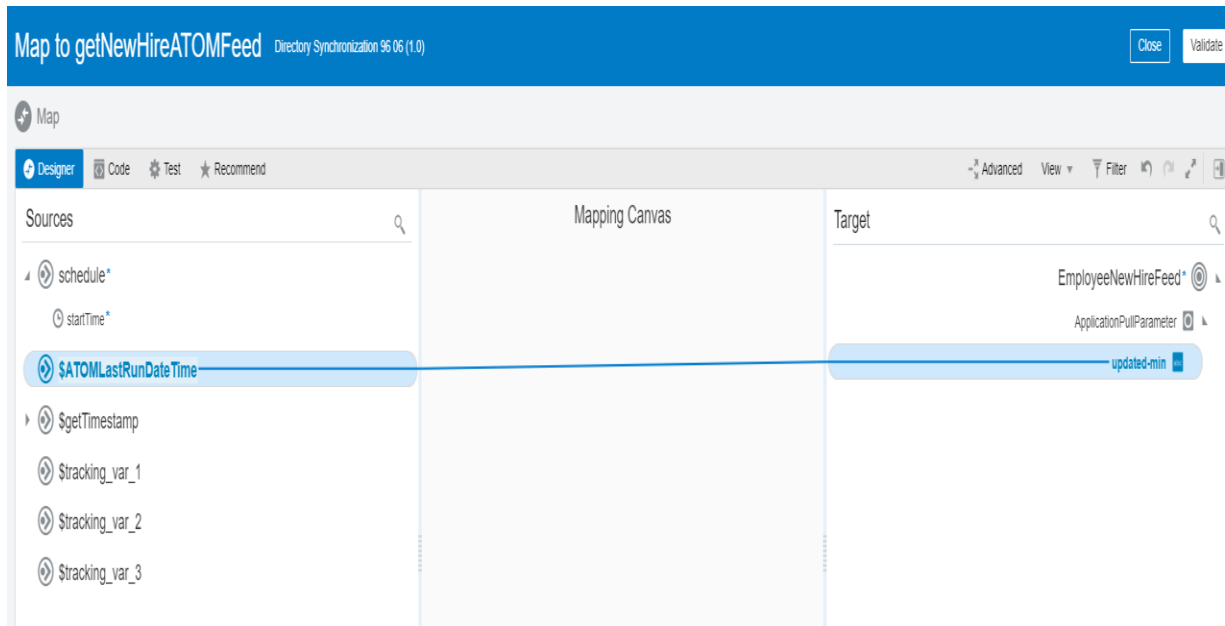
9. Click **Next**. This will display the **Summary** tab.
10. Click **Done**, this will return you to the integration canvas.
11. Click on Save



### Map to getNewHireATOMFeed

A Map to icon will now be displayed between the **getTimestamp** and **getNewHireATOMFeed**. This represents the input values parameters for the ATOM Feed just added.

1. Click the Mapper icon (**Map to getNewHireATOMFeed**).
2. Click **Edit** to invoke the mapper.
3. Select the **\$ATOMLastRunDateTime** from the **Sources** section and drag it on to the **updated-min Target** field.



4. Click Validate and Click on **Close**. This will return you to the integration canvas.
5. Click on Save

### countOfNewHires

The next task in the integration is to count the number of new hires returned in the ATOM feed. By default the ATOM feed does not return the total number of new hires, so OIC needs to be able to derive this. This is achieved by adding an **Assign Action** to the integration.

1. In the right navigation pane, click **Actions**
2. Drag the **Assign** icon to the + sign following the **getNewHireATOMFeed** label.
3. Enter "countOfNewHires" for the Assign action when prompted, Click **Create**. This will open the **Expression** page configuration

Create Action

Assign

Please give a unique name and description to this action.

Name

countOfNewHires

Description

Enter description

Create

Cancel

8. Click the + icon.

9. Enter countOfNewHires\_assignment\_1 as the **variable name**, Count the # of new employees returned as the optional **Description**

ORACLE Integration Cloud

countOfNewHires Directory Synchronization (1.0.1) Close Validate

**Assign**

Assign variables to your integration. You can assign values to variables using the editor. Variable assignments can be a greater of complexity. For example, you can use assignments in other activities and in maps.

Add at least one named variable and specify its value by adding an expression.

| Variable                         | Data Type | Description                           | Operation | Value             |
|----------------------------------|-----------|---------------------------------------|-----------|-------------------|
| (x) countOfNewHires_assignment_1 | simple    | Count the # of new employees returned |           | Add an expression |

+

10. Select the pencil to edit the **Value**. This will open the **Expression** builder.
11. Select and drag the **Count** Function (Available under Node-set) into the **Expression** text box.
12. Select the **EmployeeNewFeedResponse** value from the **Source** component and drag into **Count** expression within the **Expression** text box.

**Inputs**

View Filter Detach

**Source** Find...

- <> \*schedule
- <> \*startTime
- \$ATOMLastRunDateTime
- <> \$getNewHireATOMFeed
  - <> \*EmployeeNewHireFeedResponse
  - <> \*EmployeeNewHireFeed\_Update
- <> \$getTimestamp
- <> \*output\_getTimestamp\_dt

**Expression**

count(\$getNewHireATOMFeed/nsmpr2:EmployeeNewHireFeedResponse)

**Expression Summary**

count( EmployeeNewHireFeedResponse)

13. Click **Validate** in the title bar to validate the Expression.
14. Click **Close**. This will return you to the **Parameter** configuration page.

| Variable                         | Data Type | Description                           | Operation | Value                               |
|----------------------------------|-----------|---------------------------------------|-----------|-------------------------------------|
| (x) countOfNewHires_assignment_1 | simple    | Count the # of new employees returned |           | count( EmployeeNewHireFeedResponse) |

15. Click **Validate** and Click on **Close**. This will return you to the integration canvas.
16. Click on **Save**

## Switch Action

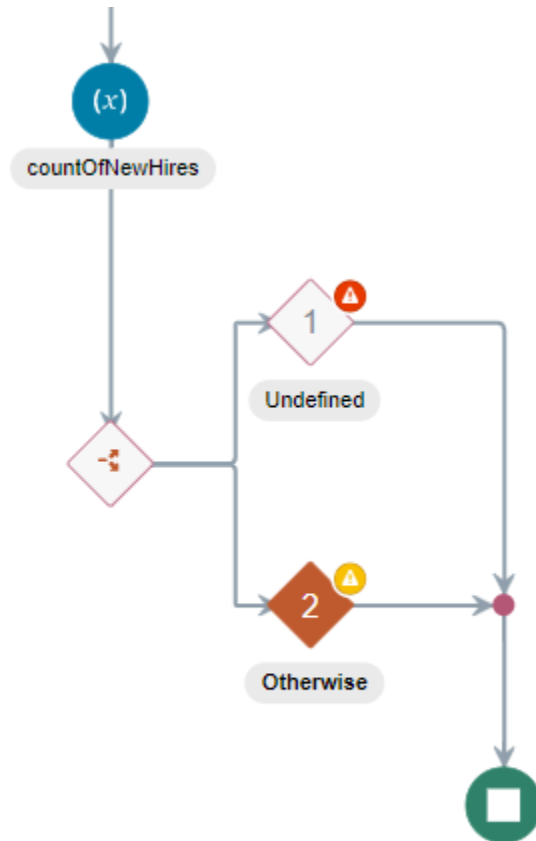
At this point in the integration it is necessary to identify whether there are any new records returned in the ATOM feed. This is achieved by using a **Switch** action.

- On the right side of the integration canvas, click **Actions** to expand the panel.
- Drag the **Switch** icon to the integration canvas to the + sign following the **countOfNewHires** label.



3. Two branches are automatically created:

- **Undefined** (first) branch: This is the branch where the **countOfNewHires** will be validated for routing
- **Otherwise** (second) branch: This branch is taken if the routing expression for the initial branch does not resolve to true.



4. Click the **Undefined** branch icon.
5. Select the **Edit** icon from the menu that is displayed. This invokes the **Expression Builder**.
6. From the **Source** component, select the **\$countofNewHires\_assignment\_1** and click on > symbol to move it to the first text box for the new condition.

## Expression in "Directory Synchronization 96 06 (1.0)"

The screenshot displays the Oracle Product Management interface for creating an expression. On the left, the 'Inputs' section shows a list of sources. The source '\$countOfNewHires\_assignment\_1' is highlighted with a red box. On the right, the 'Expression Name' section shows a 'New Condition' box. The expression name '\$countOfNewHires\_assignment\_1' is entered in the text box, and the dropdown menu is set to '='. A red box highlights the 'New Condition' box. At the bottom right, a red box highlights a button with a right-pointing arrow.

7. Select = as value in the drop down

8. Enter 0.0 is the second text box condition.

This will result in the routing of the integration through this condition if there are no new hires returned within the ATOM feed.

Expression in "Directory Synchronization 96 06 (1.0)"

**Inputs**

View ▾ Filter Detach

**Source** Find...

- ◀ <> \*schedule
- <> \*startTime
- \$ATOMLastRunDateTime
- \$countOfNewHires\_assignment\_1**
- ◀ <> \$getNewHireATOMFeed
  - ◀ <> \*EmployeeNewHireFeedResponse
    - ▶ <> \*EmployeeNewHireFeed\_Update
- ◀ <> \$getTimestamp
  - <> \*output\_getTimestamp\_dt
- \$tracking\_var\_1
- \$tracking\_var\_2
- \$tracking\_var\_3

**Expression Name**

Provide a name or short description for the expression.

New Condition + >>

\$countOfNewHires\_assignment\_1

= ▾

0.0

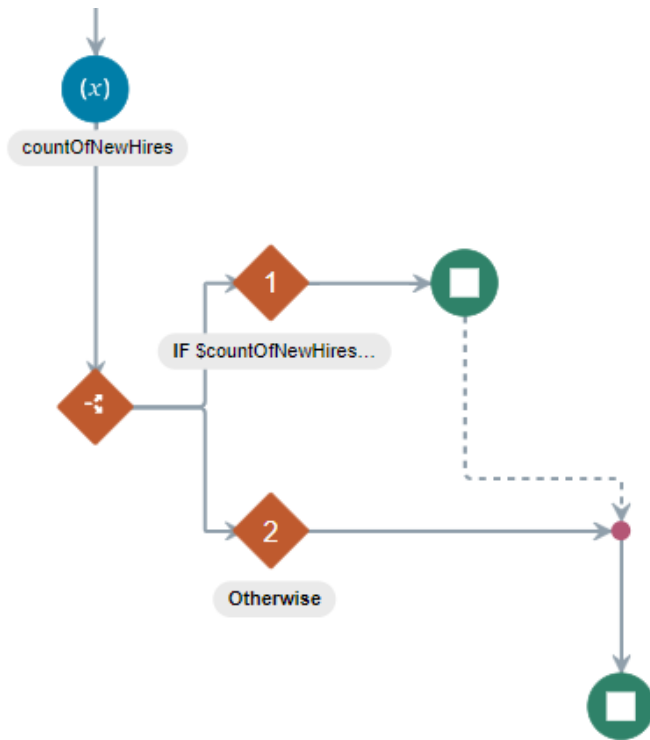
- Click on Validate and Click Close. This displays the integration canvas
- Click on Save

### Add Stop Action

In order for the integration to complete when there are no new hires present in the ATOM Feed file, it is necessary to include a **Stop** action. This is achieved in the following way.

- On the right side of the integration canvas, click **Actions** to expand the panel.
- Drag the **Stop** icon to the integration canvas to the + sign following the **IF \$countOfNewHires** label.

This will add a new stop action to the canvas as shown below.



### StageFileRef (Assign Activity)

The next task in the integration is to create a temporary variable to hold the file reference of a stage file activity, which we are going to perform in the next section.

1. Drag and drop Assign activity after Otherwise activity and enter the name as “StageFileRef”
2. Click on + symbol to create a variable and assign value as “”

StageFileRef
Directory Synchronization 96.06 (1.0)
Close
Validate

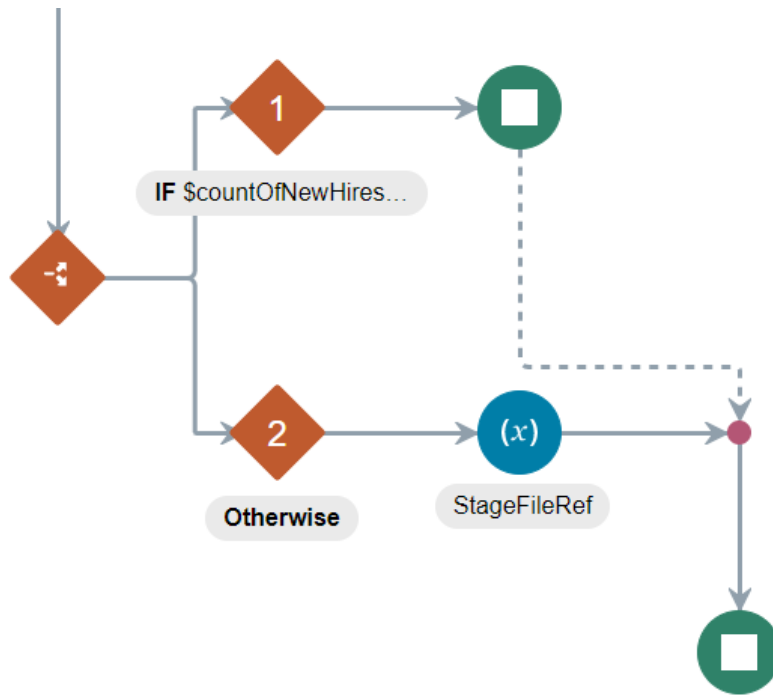
Assign

Assign variables to your integration. You can assign values to variables using the editor. Variable assignments can be a greater of complexity. For example, you can use assignments in other activities and in maps.

Add at least one named variable and specify its value by adding an expression.

| Variable                      | Data Type | Description        | Operation | Value |
|-------------------------------|-----------|--------------------|-----------|-------|
| (x) StageFileRef_assignment_1 | string    | Type a description |           | / "   |

3. Click on Validate and Close and Click on Save.



### ForEachEntry

In order to retrieve the additional details required by the downstream Identity Management System, it is necessary to query HCM Cloud and build a richer data set than returned via the ATOM feed.

The **For Each** action enables you to loop over a repeating element and execute one or more actions within the scope of the **For Each** action. The number of loop iterations is based on a user-selected repeating element.

The most effective approach in retrieving this data is via the HCM Cloud REST service. The next action to be added to the integration canvas is a **ForEach Loop** to cycle through each individual new hire in the returned ATOM feed.

1. On the right side of the integration canvas, click **Actions** to expand the panel.
2. Drag the **For Each** icon to the integration canvas to the + sign following the **Otherwise** label. This will add a new **For Each** icon to the canvas and open the configure **Create Action** dialogue window.
3. Enter ForEachEntry as the Name of the action, refer the below screenshot in point number 5
4. Select the **EmployeeNewHireFeed\_Update** element from the source component and click on > to move into the **Repeating Element**, refer the below screenshot in point number 5
5. Enter CurEntry as the **Current Element Name** value, refer the below screenshot  
The **EmployeeNewHireFeed\_Update** is the parent level within the JSON ATOM Feed result file which represents an individual new hire. The **For Each** action will loop through the JSON file until each individual new hire has been processed.

## Create Action



## For Each

View Filter Detach

Source Find...

- schedule
- startTime
- \$ATOMLastRunDateTime
- \$countOfNewHires\_assignment\_1
- \$getNewHireATOMFeed
- EmployeeNewHireFeedResponse
- EmployeeNewHireFeed\_Update**
- \$getTimestamp
- output\_getTimestamp\_dt
- \$tracking\_var\_1
- \$tracking\_var\_2
- \$tracking\_var\_3

For Each action enables iterating over a repeating element and executing one or more activities within its contained scope. Specify the repeating element by selecting from the source tree on the left and provide a name for the current element. The current element can be used for activities within the For Each scope. Select Process items in parallel if elements required to be processed concurrently.

Name ForEachEntry

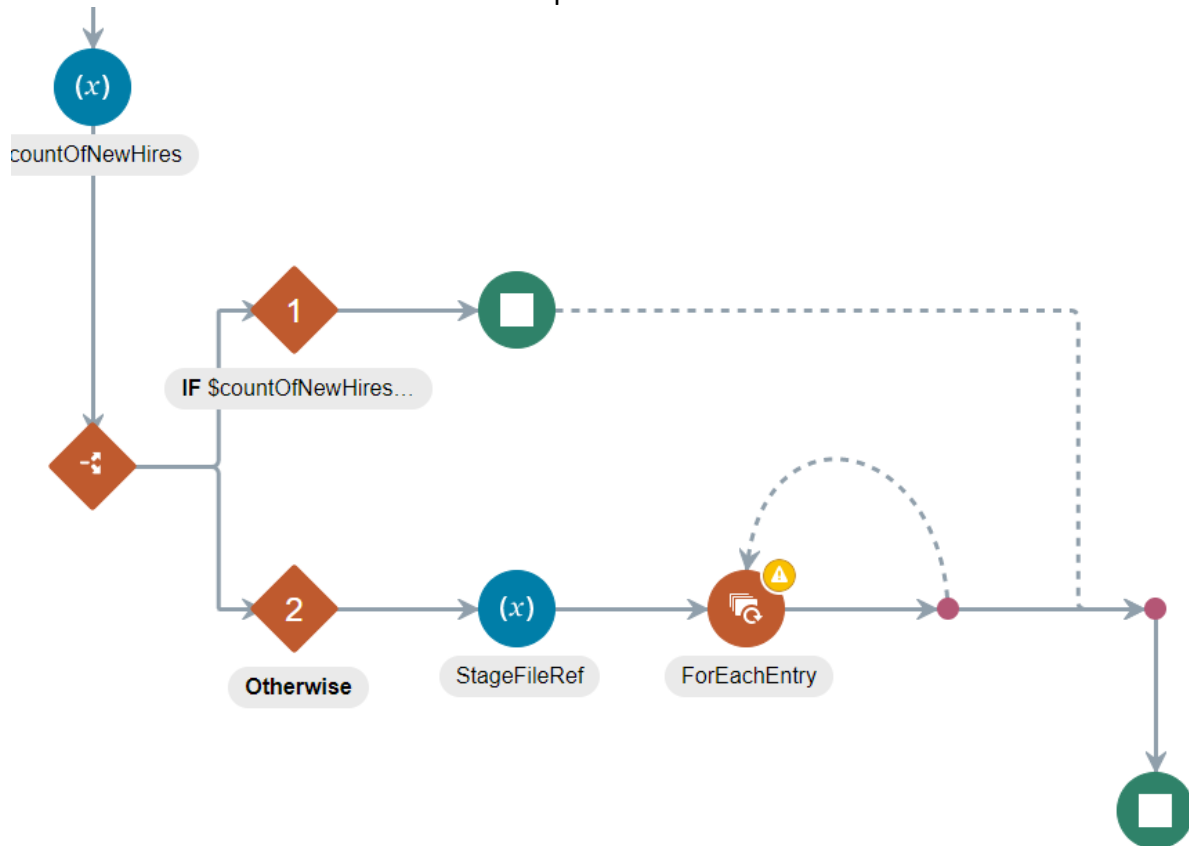
Description Enter a brief description...

Repeating Element \$getNewHireATOMFeed/nsmpr2:EmployeeNewHireFeedResponse/nsmpr2:EmployeeNewHireFeed\_Update

Current Element Name CurEntry

Process items in parallel ☐

- Click **Create**. This will close the **Create Action** dialogue window updating the integration canvas to reflect the addition of the For Each Loop.

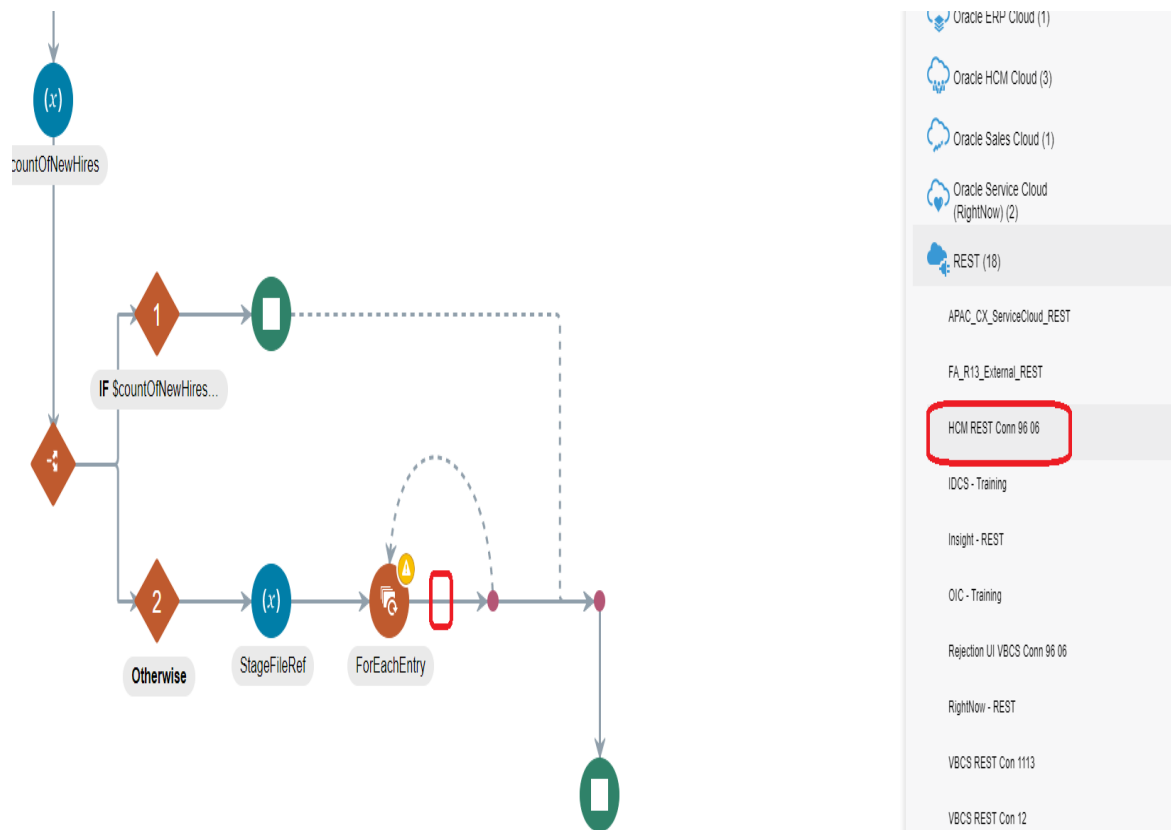


- Click on Save

### getEmployeeDetails (REST Adapter)

The next task in the integration is to make a REST call to HCM Cloud and retrieve the new hire details.

4. In the right navigation pane, click **Invokes**.
5. Click **REST**, then drag the HCM REST Conn 96 06 connection to the + sign following the **ForEachEntry** action and before the end of the arrow returning to the **ForEachEntry** Label. This ensures the call is repeated for each new hire.



6. The **REST Adapter** configuration wizard is displayed.
7. Enter getEmployeeDetails in the **What do you want to call your endpoint?**
8. Enter /emps in the **What is the endpoint's relative resource URL?**
9. Select **GET** as the value for **What action does the endpoint perform?**
10. Check **Add and review parameters for this endpoint**
11. Check **Configure this endpoint to receive the response**

**Configure REST Endpoint**

Welcome to the REST Endpoint Configuration Wizard  
This wizard helps you configure an endpoint using the REST adapter.

**Basic Info**

Request Parameters

Request

Request Headers

Response

Response Headers

Summary

\* What do you want to call your endpoint?

getEmployeeDetails

What does this endpoint do?

Describe the endpoint's purpose and detail

\* What is the endpoint's relative resource URI?

/emps

\* What action do you want to perform on the endpoint?

GET

Based on your selections, you can add parameters or configure a request and/or response for this endpoint.

Select any options that you want to configure:

☒ Add and review parameters for this endpoint

☐ Configure a request payload for this endpoint


☒ Configure this endpoint to receive the response

12. Click **Next**. This will display **Request Parameters** tab.
13. Enter q in the **Name** field. This represents the value that will be used as the input query for the REST call.
14. Select **string** as the value for **Data Type** drop down



**Configure REST Endpoint**


Help ▾ < Back Next > Cancel Done

 **Configure the Request Query Parameters**  
Configure the request query parameters for this endpoint.

**Basic Info** **Request Parameters** Request Request Headers Response Response Headers Summary

\* Resource URI /emps

Specify Query Parameters

 Detach + X

| Name | Data Type |
|------|-----------|
| q    | string    |

Template Parameters

Displays the template parameters in the relative resource URI. Template parameters are determined by details you specified

15. Click **Next**. This will display **Response** tab.
16. Select **JSON Sample** as the value for **Select the response payload format** drop down
17. Click <<<inline>>> next to the **-OR- enter sample JSON label**. This will display **Response Sample Json Payload** tab.
18. Copy the contents of the **getEmployeeResponse.json** file into the text box. This file is part of the delivered artifacts with this guide.
19. Select **OK**. This will return the display to the **Response** tab.
20. Select the **JSON** radial value for the **Select the type of payload with which you want the endpoint to reply**.

**Configure REST Endpoint**

Help ▾ < Back Next > Cancel Done

**Configure the Response Payload**  
Specify the response payload details for this integration.

**Basic Info**  
Request Parameters  
Request  
Request Headers  
**Response**  
Response Headers  
Summary

Select the attachment processing options

☐ Process attachments from response

☐ Response is HTML form

Select the response payload format

JSON Sample ▾

? Sample Location  No file chosen --OR-- enter sample JSON <<< inline >>>

\* Element

Select the type of payload with which you want the endpoint to reply

☐ XML

☐ XML(text)

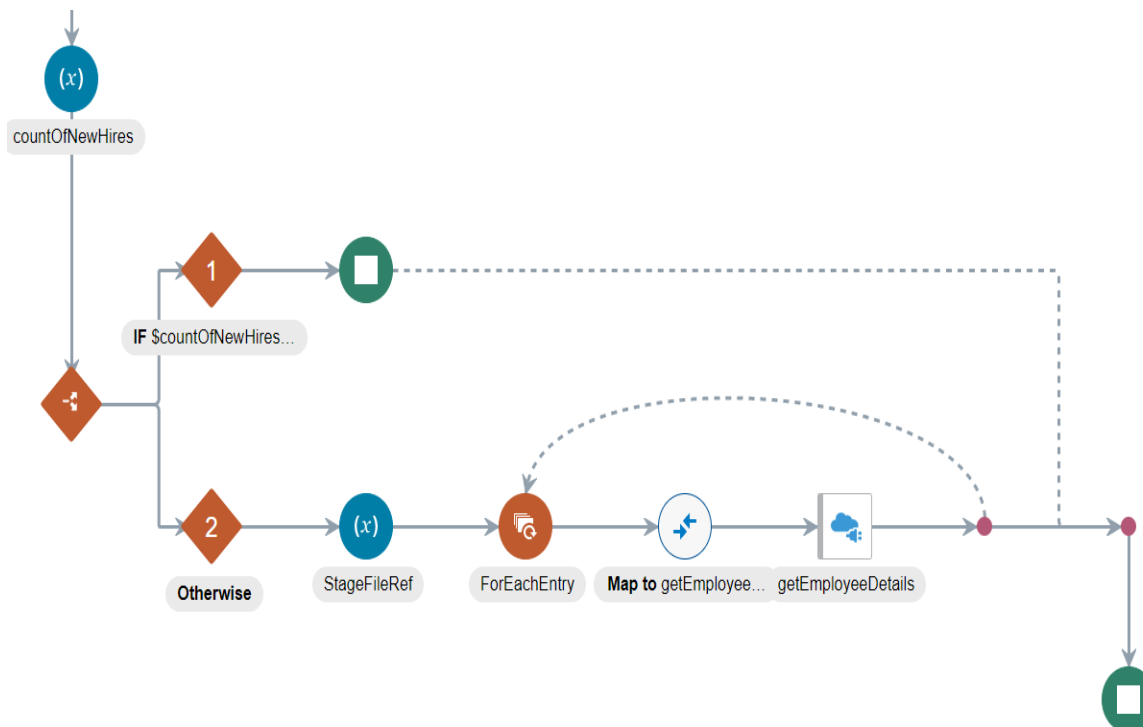
☒ JSON

☐ Other Media Type

? Media Type

21. Click **Next**. This will display **Summary** tab.

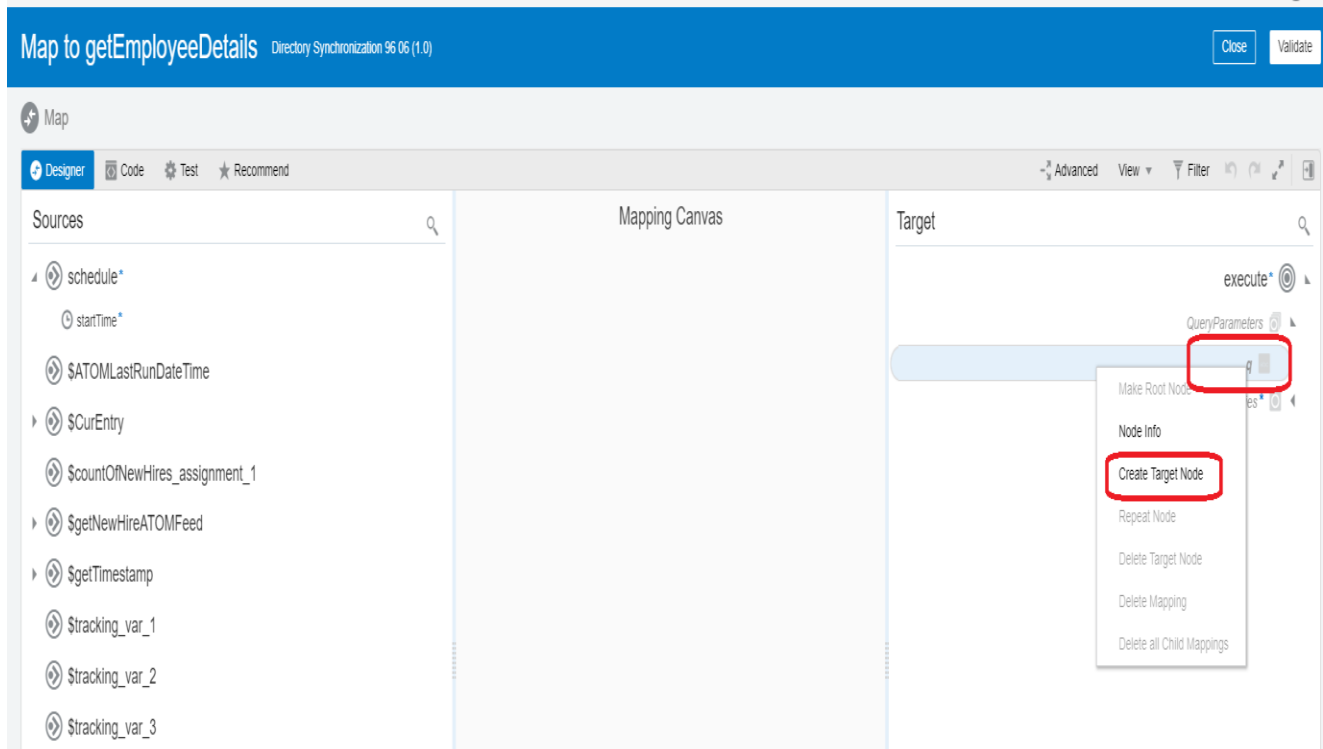
22. Select **Done**. This will add both **getEmployeeDetails** REST Adapter and **Map to getEmployeeDetails** icons to the integration.




## Map to getEmployeeDetails

Next it is necessary to define the inputs for the **getEmployeeDetails** step.

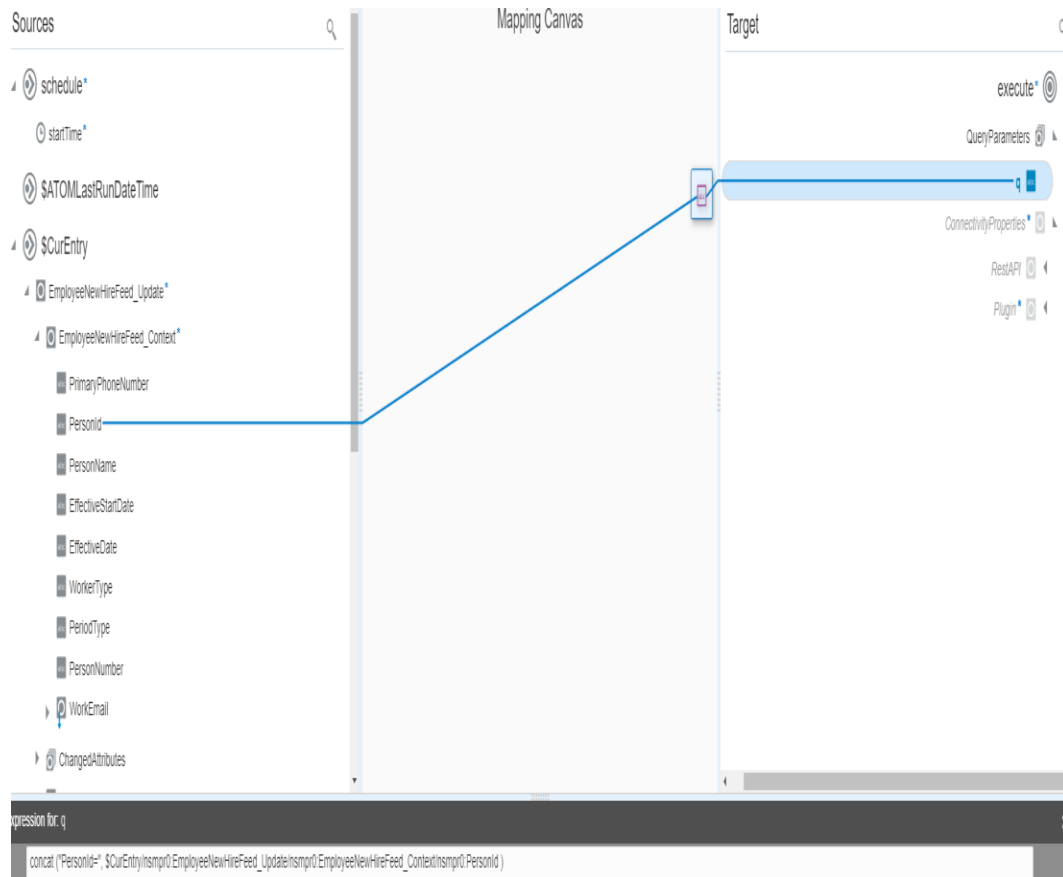
1. Click the Mapper icon (**Map to getEmployeeDetails**).
2. Click **Edit** to invoke the mapper.
3. Right click on **q** which is under execute → QueryParameters on the **Target** section. And click on Create Target Node. This will open the **Expression for: q** panel in the lower part of the UI.



4. Select from the **Toggle Functions**  icon to expose the available **Components** and **Operators**.
5. Select and drag the **string concat** field into the **Expression for : q** panel.
6. Enter "PersonId=" as the first line value
7. Select PersonId from CurEntry → EmployeeNewHireFeed\_Update → EmployeeNewHireFeed\_Context and drag into the second line value under the **concat** statement. At runtime this statement will concat the value of PersonId= with the actual value returned in the **PersonId** field for a new hire. This will construct the query value for the REST call made to HCM Cloud.

The REST call being constructed will look like the following where [PersonIdValue] is the value returned from the ATOM feed for an individual Employee.

[https://<hostName>/hcmCoreApi/resources/latest/emps?q=PersonId=\[PersonIdValue\]](https://<hostName>/hcmCoreApi/resources/latest/emps?q=PersonId=[PersonIdValue])



8. Select **Validate** and **Close**. This will display the integration showing a configured **Map to getEmployeeDetails** icon.
9. Click on Save to save Integration flow.


### writeNewUserRecord to Staging

The next task in the integration is to write the new user record into the staging file using the Stage activity.

1. In the right navigation pane, click **Actions** and drag and drop Stage file activity after getEmployeeDetails.
2. Enter the "WriteRecordToStage" in **What do you want to call your action** and click on Next

**Configure Stage File Action**

Help ▾ < Back Next > Cancel Done

 **Welcome to the Stage File Action Configuration Wizard** X  
This wizard helps you to configure Stage File Action. You will be asked to define parameters and schema for performing the file operation.

**Basic Info**

Configure Operation

Schema Options

Format Definition

Summary

\* What do

WriteRecordToStage

What does this action do?


Describe the action's purpose and detail

Provide a meaningful name so that others can understand the action. You can include English alphabetic characters, numbers, underscores and dashes. You can not include blank spaces, special characters and multibyte characters.

- Choose Stage File Operation as “Write File” and Specify the file name as “newCandidateRecord.csv” and specify the Output Directory as “/tmp” and check the Append to Exist file option and click Next

**Configure Stage File Action**

Help ▾ < Back Next > Cancel Done

 **Configure the Stage File Action Parameters for the Selected Operation** X  
Define the parameters for Stage File Operation

Basic Info


**Configure Operation**


Schema Options

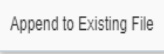
Format Definition

Summary

? \* Choose Stage File Operation Write File ▾

\* Specify the File Name "newCandidateRecord.csv" 


\* Specify the Output Directory "/tmp" 

Append to Existing File ☒ 

- Keep the default options as it is and click Next

**Configure Stage File Action**

Help ▾ < Back Next > Cancel Done

 **Configure the Stage File Action with Schema Options** ×  
Define the schema options for Stage File Operation

☒ Basic Info  
☒ Configure Operation  
☒ **Schema Options**  
 Format Definition  
 Summary


? Do you want to specify the structure for the contents of the file?  
☒ Yes ☐ No

? Do you want to create a new schema or select an existing one?  
☒ Create a new schema from a CSV file  
☐ Select an existing schema from the file system

- Click on “Choose File” button and select the file newEmployeeFile.csv which is given along with this document. The structure of the csv will be displayed in the lower half of the page.

**Configure Stage File Action**

Help ▾ < Back Next > Cancel Done

 **Define the Schema Format** ×  
Define the parameters for generating an XSD for native format or non-XML languages. The XSD is used at runtime to translate a native format message into an XML message and vice versa.

☒ Basic Info  
☒ Configure Operation  
☒ Schema Options  
☒ **Format Definition**  
 Summary

Create a New Schema from a CSV file

Select a New Delimited Data File **Choose File** No file chosen

Selected File Name newEmployeeFile.csv

\* Enter the Record Name

\* Enter the Recordset Name

Select the Field Delimiter

Character Set

Optionally Enclosed By

☒ Detach   
 Use First Row as Column Headers ☒   
 Mark All As Optional ☐

| Salutation  | FirstName   | LastName    | CorrespondenceLan | PersonN |
|-------------|-------------|-------------|-------------------|---------|
| String ▾    | String ▾    | String ▾    | String ▾          | String  |
| Mandatory ▾ | Mandatory ▾ | Mandatory ▾ | Mandatory ▾       | Mandato |

- Enter User value for the Enter the **Record Name** field.
- Enter NewUser for the Enter the **Recordset Name** field.
- Select **Comma(,)** drop down value for **Select the Field Delimiter**
- Select **ASCII** drop down value for **Character Set**

10. Select “ drop down value for **Operationally Enclosed By**

**Configure Stage File Action**

Help < Back Next > Cancel Done

**Define the Schema Format**  
Define the parameters for generating an XSD for native format or non-XML languages. The XSD is used at runtime to translate a native format message into an XML message and vice versa.

**Basic Info**  
**Configure Operation**  
**Schema Options**  
**Format Definition**  
Summary

Create a New Schema from a CSV file

Select a New Delimited Data File  No file chosen

Selected File Name newEmployeeFile.csv

\* Enter the Record Name User

\* Enter the Recordset Name NewUser

Select the Field Delimiter Comma (,) ▾

Character Set ASCII ▾

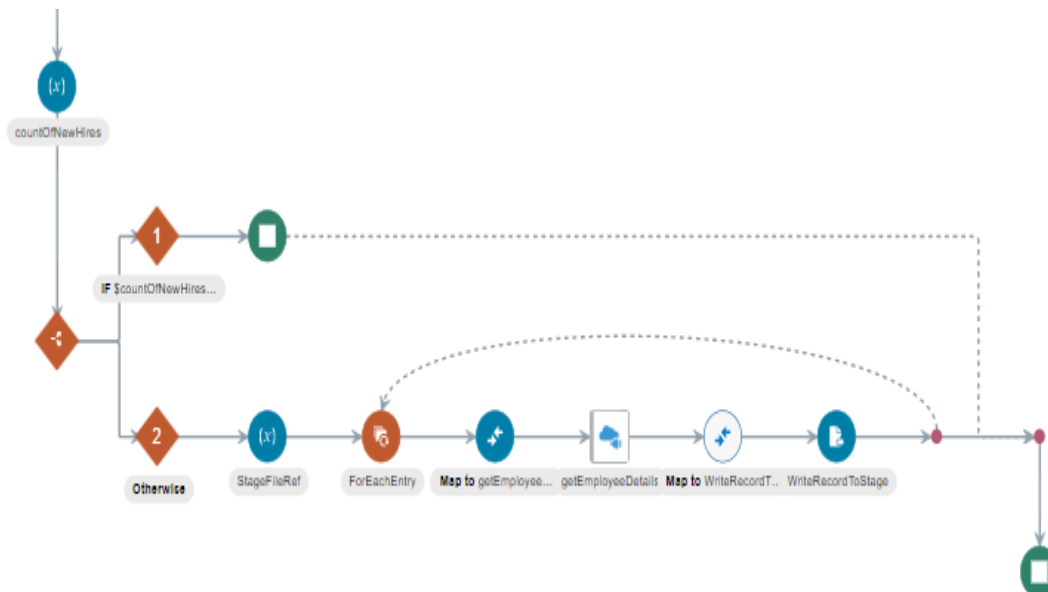
Optionally Enclosed By " ▾

☒ Detach Use First Row as Column Headers ☒ Mark All As Optional ☐

| Salutation  | FirstName   | LastName    | CorrespondenceLan | PersonN |
|-------------|-------------|-------------|-------------------|---------|
| String ▾    | String ▾    | String ▾    | String ▾          | String  |
| Mandatory ▾ | Mandatory ▾ | Mandatory ▾ | Mandatory ▾       | Mandato |

No Configuration.

11. Click **Next**. This will display **Summary** tab and click on Done. This will return you to the integration canvas.



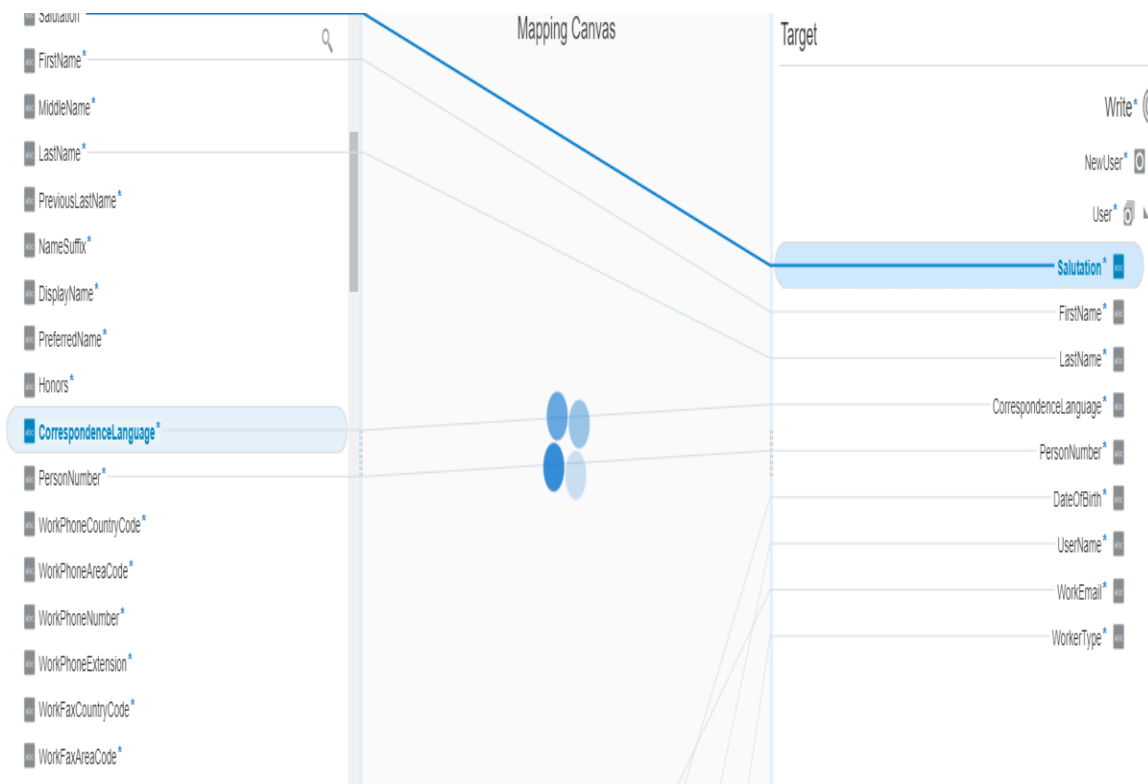
12. Click on Save

### Map to WriteRecordToStage

This map to action will represent the transformation from JSON format to the CSV file format defined in the Stage action.

1. Click the Mapper icon (**Map to WriteRecordToStage**).
2. Click **Edit** to invoke the mapper.
3. Select the fields under the **getEmployeeDetails** → **executeResponse** → **response-wrapper** → **items** form the **Source** side of the UI and drag them to the corresponding **Target** values.

The **Target** values are under **NewUser** and **User**. For the purposes of this guide select the fields which will generate a meaningful output file.



4. Click on Validate, Click on Close and Click on Save on Integration Canvas.
5. Drag and drop Assign activity after “writeRecordToStage” activity and enter name as assignStageFileRef and click on Create
6. Click on + symbol and select the Variable “StageFileRef\_assignment\_1” from drop down



Assign variables to your integration. You can assign values to variables using the editor. Variable assignments can be a greater of complexity. For example, you can use assignments in other activities and in maps.

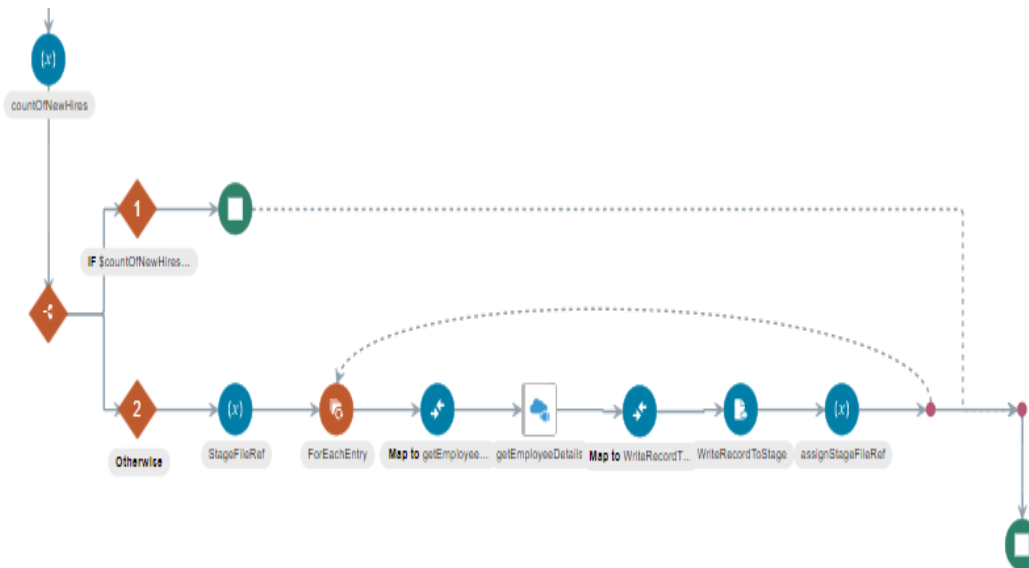
Add at least one named variable and specify its value by adding an expression.

| Variable   | Data Type | Description        | Operation | Value             |
|--|-----------|--------------------|-----------|-------------------|
| (x) assignStageFileRef_assignment_1  | string    | Type a description |           | Add an expression |
| <div> ATOMLastRunDateTime (Schedule Parameter)<br/> Schedule_Receive (Schedule Parameter)<br/> <b>StageFileRef_assignment_1</b><br/> countOfNewHires_assignment_1 </div> |           |                    |           |                   |

- Click on Expression Builder under Value
- Expand WriteRecordToStage → WriteResponse → WriteResponse → ICSFile and select FileReference and click on > symbol

The screenshot shows the Oracle Integration Cloud (OIC) interface for assigning a value to a variable. The 'Source' dropdown is expanded, showing a tree structure of variables. The path '\$WriteRecordToStage' > '\*WriteResponse' > '\*WriteResponse' > 'ICSFile' > '\*FileReference' is selected, and a blue arrow button is clicked. The 'Expression' field on the right contains the path '\$WriteRecordToStage/nsmpr5:WriteResponse/nsmpr6:WriteResponse/nsmpr3:ICSFile/nsmpr3:FileReference'.

- Click on Validate and close. Click on Validate and close and Click on Save.



### WriteStageFileToFTP (FTP Adapter)

The next task in the integration is to add the **FTP Adapter** to the integration representing the step where the Stage file is written to the customer FTP server.

13. In the right navigation pane, click **Invokes**.

14. Click **FTP**, then drag the “FTP Conn 96 06” connection to the + sign after **ForEachEntry** loop and at the end of the Otherwise section. The **FTP Adapter** configuration wizard is displayed.

15. Enter WriteStageFileToFTP in the **What do you want to call your endpoint?**

16. Click **Next**. This will display **Operations** tab.

17. Select the **Write File** value in the **Select Operation** drop down.

18. Select the **ASCII** radial selection for the **Select a Transfer Mode**

19. Enter the **Output Directory** value location based on your FTP folder structure configuration given below. Please create the directory structure if it is not there in FTP location.

/upload/public\_ftp/<<YOURNAMES>>/directsynch

20. Enter the **File Name Pattern** value. Depending on the specific requirements, the file name can follow different pattern structures. A simple example would be as follows:

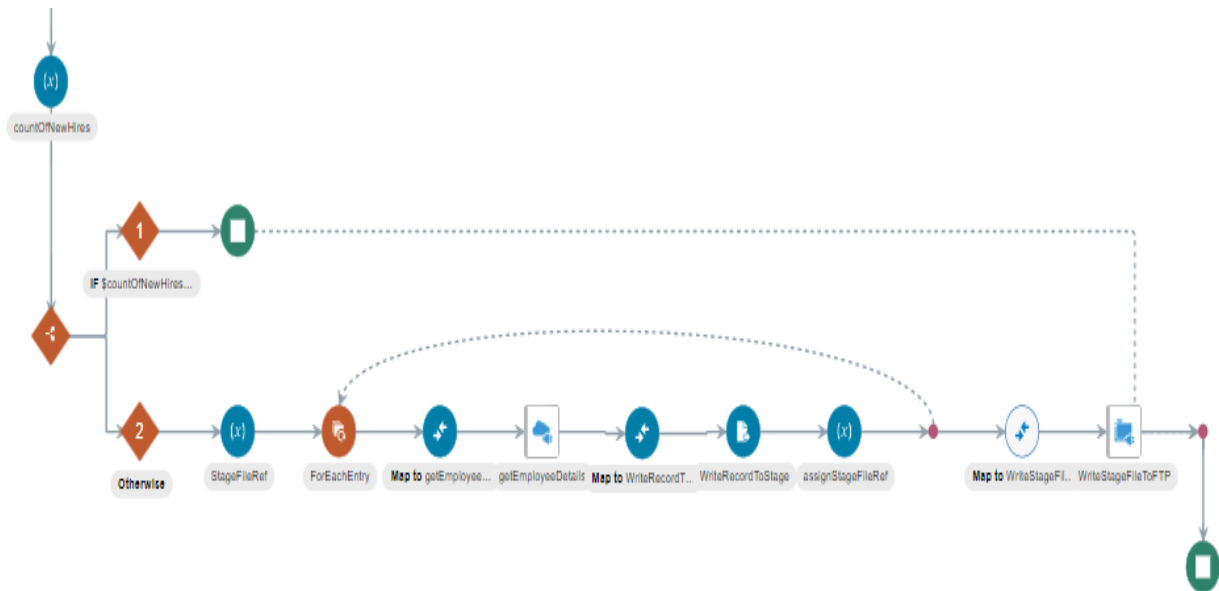
newCandidateRecord%yyyyMMddHHmmss%.csv which would result in a file named newCandidateRecord20180418082100.csv for a file written on the 18<sup>th</sup> of April 2018 at 08:21:00 AM.

21. Click **Next**. This will display **Schema** tab.

22. Select the **No** value for the **Do you want to specify the structure for the contents of the file?**

23. Click **Next**.

24. Click **Done**, returning to the integration canvas. This will add both **WriteStageFileToFTP** and a **Map to WriteStageFileToFTP** icon to the integration.



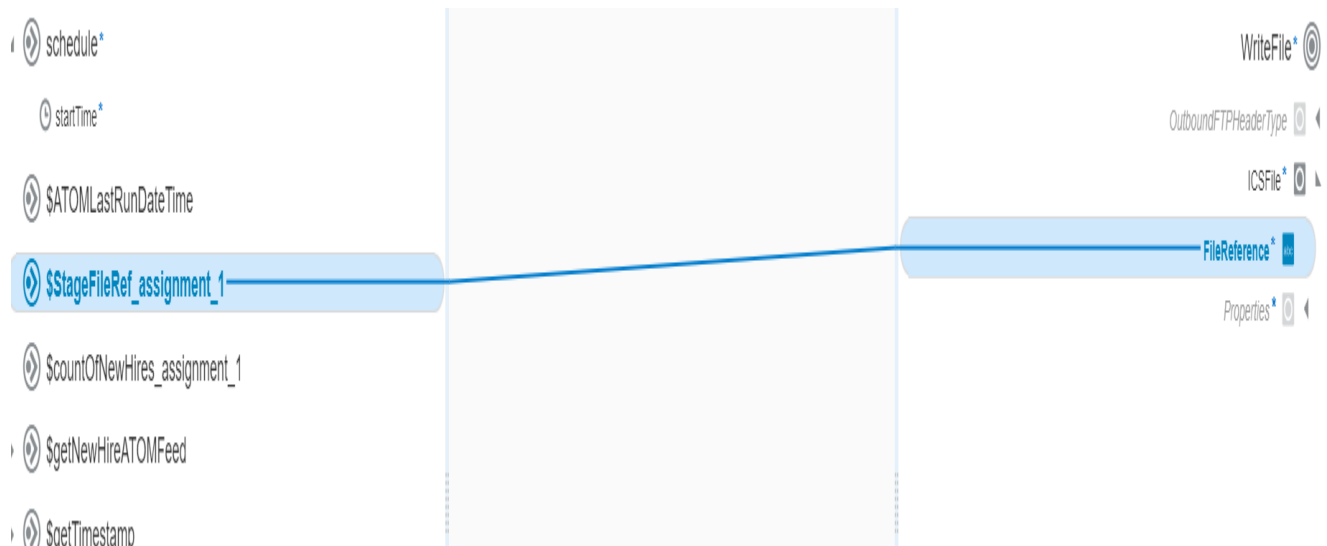
### Map to WriteStageFileToFTP

This map to action will map the file reference to FTP adapter.

10. Click the Mapper icon (**Map to WriteStageFileToFTP**).

11. Click **Edit** to invoke the mapper.


12. Select the StageFileRef\_assignment\_1 on the **Source** side of the UI and drag it to the FileReference under ICSFile on **Target**.

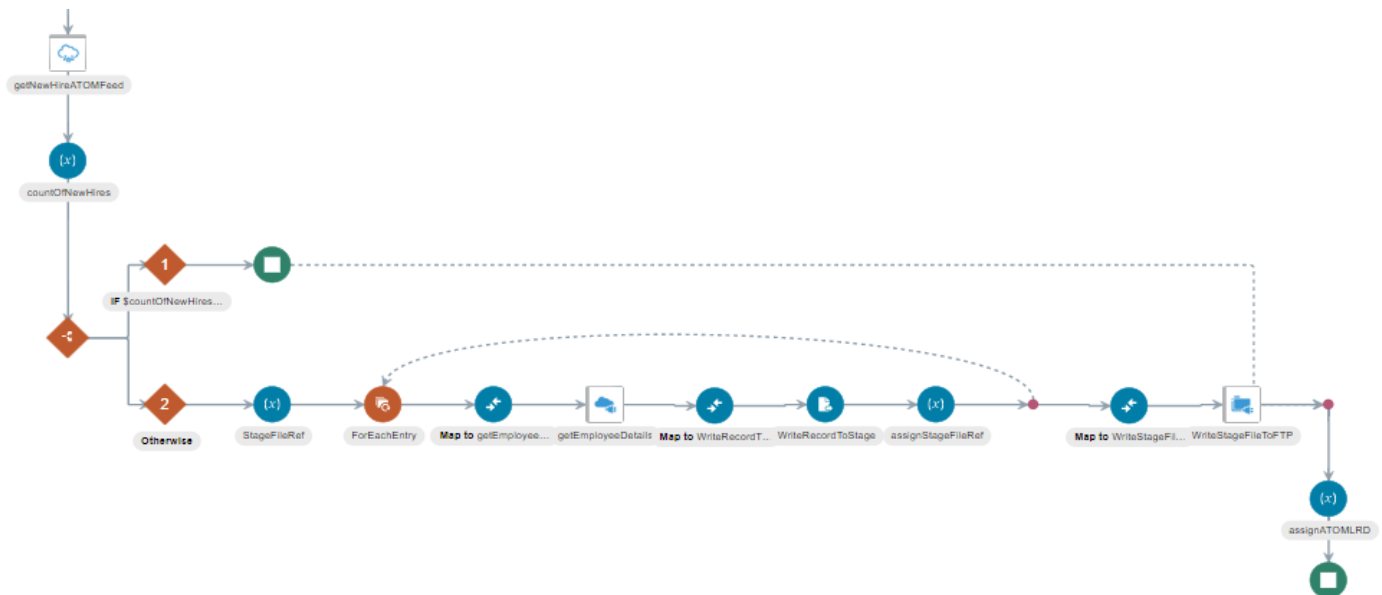


13. Click Validate and Click on **Close**. This will return you to the integration canvas. And click on Save

## assignATOMLRDT

The final step in defining the integration is to set the ATOM Last Run Date for use next time the integration is run.

1. In the right navigation pane, click **Action**
2. Drag the **Assign** icon to the + sign following the WriteStageFileToFTP (after Switch condition) and before the **Stop** icon.
3. Enter assignATOMLRD as the **Name** value for the newly created action.
4. Select **Create**. This will open the **Assign** configuration page. Click on + symbol
5. Select the **ATOMLastRunDateTime (Schedule Parameter)** in the **Variable** drop down.
6. Select the  pencil icon in the **Value** field. This will open **Expression** dialogue window.
7. Select and drag and the **OutputGetTimestamp\_dt** into the **Expression** field.
8. Click on Validate and Click on **Close**. The **Assign** configuration page will be displayed.
9. Click on Validate and Click on **Close**. This displays the integration canvas.
10. Click on Save



## Tracking

The final step in configuring the integration is the setting of the business identifier. This is used to provide a unique value for monitoring and reporting.

1. From the action menu on the integration canvas, select the **Tracking** option. This will open the **Tracking** dialogue window
2. Drag the **startTime** value from the Source panel on to the first **Tracking Field**.
3. Select **Save**. This will close the **Business Identifiers for Tracking** dialogue window and return you to the integration canvas.
4. Select **Save**. You are now ready to activate and execute the integration.

## Activate and Run the Integration.

1. Find your integration flow on the list and click the **Activation switch**. This will open the **Activate Integration** dialogue window.
2. Check the **Enable tracing**
3. Check the **Include payload**
4. Select the **Activate** button
5. Once the **Activation** is complete, select from the action menu the **Submit Now** action.
6. This will execute the integration. A **request id** value will be displayed.
7. Click the **request id** value. This will open the **Monitor Runs** page. Here you can view the execution of the integration. Once the integration has completed a Successful notification will be displayed.
8. Select the **RUN ID** value. This will open the Tracking summary page for this integration.
9. Select the **start Time:** value (which is the **Business Identifier** set on the integration earlier). This will open the integration instance screen, displaying all of the steps within the integration. They will be green, indicating that they were successful.

### **Validating Extracted File presence on FTP Server**

Following the execution of the integration within OIC, you should navigate to the folder on your FTP server to validate that the file is located there.